Nicholas Chan
nipchan@ucsc.edu
4/06/2021
CSE13s Spring '21

Design Document:
Assignment 1: LRC

Left Right Center is a game that will be played by 2-14 people for the sake of this lab. Four main actions (passing, sending $1 to the player on the left, sending $1 to the player on the right and sending $1 to the pot in the center) are determined randomly by the roll of a single dice.

In Assignment 1, I implemented a simulation of a game of Left Right Center. The main moving parts of this program are:

- Rolling a dice
    - For players with >=3 dollars, roll 3 time
    - Else, the number of rolls a player gets = the number of dollars in their bank
- The action to make once a dice is rolled
- The change in states of players (Among the starting group of players, some may hop back in and some may be forced out)

The inputs of this program are:

- A positive integer seed value to feed the srandom() function
- A positive integer from 2-14 to represent the number of players playing a game

Top Level

Pseudocode

```
Main:

    Take  seed as user input
    Take number of players as user input

    // Commence rounds until winner is found

    for ( trial when there is  >1 players with >1 in the bank)
        Player who's turn it is rolls
        for (every dollar owned by player)
            Roll dice
            If (player rolled pass)
                print (action of passing)
            Else if  (player rolled left)
                print (action of passing money to the left)
                Increment player to the left's money by 1
                Decrement this player's money by 1
            Else if  (player rolled right)
                print (action of passing money to the right)
                Increment player to the right's money by 1
                Decrement this player's money by 1
            Else if  (player rolled center)
                print (action of passing money to the left)
                Increment money in center by 1
                Decrement this player's money by 1

    for (player with >1 in their bank)
        print(winner)
```

Functions/Conditions Needed

To simulate simulate a game of LRC, features that could emulate the actions of

1. Allotting the proper starting amount of dollars to players
2. Finding the indices of players to the left or right of a player
    a. . Thankfully such functions were already given in the asgn1 instructions pdf.
3. Rolling the dice
4. Deciding a player action after a dice was rolled
5. Altering the values of dollars owned by a player
6. Naming a winner at the end of the game

would be needed for proper implementation.

Design Process

Starting this assignment, I was getting familiar with c syntax and the game of LRC. I took into account the hint that an infinite loop with an exit condition should have been used for this lab and worked on a way to implement it. Each iteration of the infinite loop is essentially one round of LRC. The four conditionals inside the

- I first realized the global variables/parameters I needed were an array of players/philosophers, an array representing the money held by active players (active players are represented by the int variable “aplayers” in my code), a max number of players and a max number of dollars that players start out with
    - To handle the possibility of players leaving or rejoining the match based on their amounts of dollars owned, I separated players by active players (int active players), starting players (int starting players) and the default pool of players we were given with the philosophers array.
    - These were all defined outside of the main function
    - The use of arrays, left and right functions and the conditional statements for player actions after a dice roll worked together to satisfy the features of allotting the proper starting amount of dollars to players and altering the values of dollars owned by a player.
- Second , I needed to take two user inputs before entering the infinite loop
    - Input 1: Seed
    - Input 2: Number of players (2-14)
- Third I needed to set the dollar amounts of all players who didn’t start the game to 0
    - This was done with a for loop before entering the main infinite loop
    - The feature previously described about declaring the number for starting players and active players would make sure that players who did not start the game never enter it.

        - The exclusion of non starting players is better described in a few bullet points.
- Once all inputs were entered, the infinite loop would have everything it needed to simulate a game of LRC.
    - Each iteration (round) consists of a player rolling the dice once for each dollar that they own, however the maximum rolls are 3.
    - States of players, player dollar amounts and the center/pot's dollar amount are updated with each iteration of the infinite loop.
    - These two variables would be incremented every round
        - uint32_t trial
            - Used to count the total number of passed trials required for the game
            - Incremented by 1 for every passing of the for loop
        - uint32_t turn
            - Used for deciding whose turn it is to roll among the starting players
            - turn is calculated from trial % splayers
                - (The trial number) %(the number of starting players)
                - This is required because the index produced by this expression would never allow players who were not a starting player to hop back into the match if their dollar amount became >0 after previously having an amount of 0
- The infinite loop ends when there is one player left with >0 dollars in their bank
- A final loop after the infinite loop checks through the bank array based on the indices 0-13 to find the index with a non-zero amount
    - This is done because there are a maximum of 14 players/philosophers in the game, and only the winner will be left with a non-zero amount in the bank array